Semi-Supervised Learning for Anomaly Detection in Blockchain-based Supply Chains

Do Hai Son*, Bui Duc Manh†, Tran Viet Khoa‡, Nguyen Linh Trung§,
Dinh Thai Hoang†, Hoang Trong Minh¶, Yibeltal Alem‡, and Le Quang Minh*
* VNU Information Technology Institute, Hanoi, Vietnam.
† School of Electrical and Data Engineering, University of Technology Sydney, Australia.
‡ University of Canberra, Australia.
§ AVITECH, VNU University of Engineering and Technology, Hanoi, Vietnam.
¶ Posts and Telecommunications Institute of Technology, Vietnam.

Abstract—**Blockchain-based supply chain (BSC) systems have tremendously been developed recently and can play an important role in our society in the future. In this study, we develop an anomaly detection model for BSC systems. Our proposed model can detect cyber-attacks at various levels, including the network layer, consensus layer, and beyond, by analyzing only the traffic data at the network layer. To do this, we first build a BSC system at our laboratory to perform experiments and collect datasets. We then propose a novel semi-supervised DAE-MLP (Deep AutoEncoder-Multilayer Perceptron) that combines the advantages of supervised and unsupervised learning to detect anomalies in BSC systems. The experimental results demonstrate the effectiveness of our model for anomaly detection within BSCs, achieving a detection accuracy of 96.5%. Moreover, DAE-MLP can effectively detect new attacks by improving the F1-score up to 33.1% after updating the MLP component.**

Index Terms—**Anomaly detection, blockchain, supply chain, autoencoder, machine learning.**

I. INTRODUCTION

Blockchain technology has experienced significant growth across various sectors, including business, finance, and beyond [1]. Since the launch of the Ethereum network in 2016, blockchain's application has expanded well beyond cryptocurrency. Nowadays, blockchain shows its impacts in many real-world applications, e.g., smart health, smart cities, and so on [1]. This is largely due to its advanced properties, including immutability, transparency, and fault tolerance. However, despite these advantages, blockchain systems are not without vulnerabilities. These weaknesses can lead to severe consequences, such as financial theft and information breaches [2].

In this work, we consider a rising application of blockchain technology, i.e., supply chain management. The blockchain-based supply chain, management, utilizes smart contracts (SMs) to record various processes including development, transportation, and consumption of commodities, such as agricultural products [3]. Although BSCs offer numerous advantages, the inherent vulnerabilities of blockchain technology can lead to significant issues, such as counterfeit goods deceiving consumers [4]. To deal with this problem, general intrusion detection system (IDS) techniques are applied to supply chain

systems. In [5], the authors pointed out that IDS can be implemented at various levels within BSCs, including the network, consensus, and beyond layers. It is also highlighted that deploying IDS at the network layer is crucial for the early detection of cyber-attacks on both traditional computer networks and blockchain networks. Therefore, in this paper, we focus on studying IDS methods to detect cyber-attacks at the network layer for BSCs.

Many studies have been implemented to develop IDS for supply chains on traditional computer networks [6]. In these works, deep learning (DL) is a state-of-the-art technique in terms of cyber-attack detection. Because supervised learning needs to be trained by known cyber-attacks, DL-based anomaly detection can be classified into two methods, i.e., unsupervised learning and semi-supervised learning [7]. On the one hand, unsupervised learning uses unlabeled datasets to train the neural network (NN). The advantage of unsupervised learning is that it can detect unknown cyber-attacks without knowing their characteristics [7]. In [8], the authors proposed a generative adversarial network (GAN)-based IDS to detect cyber-attacks on traditional computer networks. This approach can achieve accuracy up to 97% for detecting the distributed denial-of-service (DDoS) in a well-known dataset. On the other hand, semi-supervised learning uses a part of labeled datasets to train an NN to enhance the performance for the anomaly detection of unsupervised learning. In [9], the authors proposed a cascade structure of an auto-encoder (AE) and a deep neural network (DNN) to detect cyber-attacks in the NSL-KDD dataset. This network can classify known cyber-attacks in the dataset instead of just detecting abnormal network behaviors. However, the above studies are geared toward cyber-attacks on traditional computer networks and are not yet related to blockchain-based systems.

From the perspectives of IDS for blockchain-based systems, other works tried to use the above DL approaches for blockchain networks. In [10], 11 features of the Bitcoin network, e.g., *version, average, max size*, and *cost of messages*, are extracted to detect two types of blockchain cyber-attacks in the network layer, i.e., Eclipse and DoS. The proposed security mechanism can detect perfectly trained blockchain cyber-attacks using a typical AE network. In [11], the authors used

a semi-supervised GAN approach to detect the anomaly in a public blockchain dataset for general purposes. The authors in [11] used 21 transactions (txs)-based features of the dataset, e.g., *gas price*, *gas cost*, *amount of Ethereum tokens*, and so on, to achieve accuracy up to 95% for detecting anomalies in blockchain networks. In this work, we aim to detect cyber-attacks at the network layer in BSCs.

From the above literature, anomaly detection at the network layer using DL in BSCs has to face various challenges. The first challenge is the lack of a cyber-attack dataset for BSCs. Recently, in [12], the authors introduced the first synthetic dataset for studying cyber-attacks in the Ethereum network. However, it is either not for BSC purposes or does not include cyber-attacks in the consensus and beyond layers [5]. Another challenge is the difficulty of detecting abnormal txs at the network layer without the labeled dataset. For example, the overflow and underflow (OaU) vulnerability [2], in the consensus and beyond layers, leads to adversarial txs that are indistinguishable from honest txs to the same SM just by an unsupervised network.

To tackle the first challenge, we first built a BSC in the laboratory environment to capture the dataset. Our dataset consists of various types of cyber-attacks[1]: (i) at the network layer are brute-force password (BP) and denial of service (DoS), and (ii) at the consensus and beyond layers are DoS with block gas limit (DoS_gas), OaU, and flooding of transactions (FoT). To collect traffic data on Ethereum nodes, we used the BC-ID tool in [12] and the traceability protocol in [3]. To address the second challenge, we then propose a semi-supervised network named DAE-MLP to detect cyber-attacks in BSC systems. Different from [9] where the authors used a cascade structure in their neural network, we parallel the Deep AE (DAE) component (unsupervised) and multilayer perceptron (MLP) component (supervised). The output of DAE-MLP is a combined anomaly score that is used to detect abnormal behaviors. There are two reasons for proposing this network structure. First, DAE-MLP can detect anomalies in the consensus and beyond layers with only the information from captured packets at the network layer, thanks to the labeled dataset, while retaining part of the advantage of an unsupervised learning-based detector. Second, MLP's hyper-parameters can be individually updated for novel cyber-attacks without re-training the DAE component. Our experiments show that our proposed DAE-MLP network can outperform other considered models (i.e., unsupervised learning networks and binary classifiers) with an accuracy of up to 96.5%. Additionally, DAE-MLP shows its ability to update novel cyber-attacks. F1-score of detecting new cyber-attacks before and after updating the MLP component is improved up to 33.1%.

II. BSC SYSTEMS AND OUR PROPOSED ANOMALY DETECTION MODEL

In this paper, we study blockchain-based supply chains; an example of such applications is agriculture traceability. To do
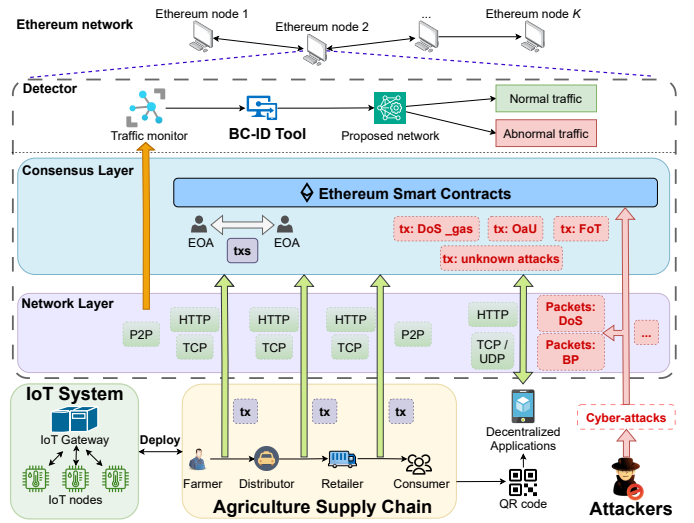
Fig. 1: Proposed system model of anomaly detection on the Ethereum-based agriculture supply chain.

this, we first briefly present the protocol of a BSC system for agriculture traceability. We then demonstrate our system model to detect abnormal traffic data in the BSC at the network layer.

A. Blockchain-based Agriculture Supply Chains

Since BSC is an emerging technology, it requires both modern hardware and communication protocols. Regarding hardware, at the farm field, BSC systems expect IoT devices, i.e., IoT sensors (e.g., humidity, temperature, motion, etc.), IoT gateways, and blockchain nodes (e.g., Ethereum). The IoT sensors read the environmental information and transmit it to IoT gateways. These values are packed into txs at the gateway and broadcast to the blockchain nodes based on BSC protocols. In this study, we use a BSC protocol that was proposed in [3]. In [3], the authors introduced an SM and storage protocols for a soybean traceability process. In detail, there are seven participating entities in the traceability process, i.e., seed company, farmer, grain elevator, grain processor, distributor, retailer, and end customer. Each entity has its role, such as a seed company having to update the following information into the traceability SM, i.e., company name, lot coordinates, seed brand, certifying agency, and variety. Following that, farmers also have to push information, e.g., product ID, field ID, chemical application, harvest data, date sole, and so on, during cultivation. The rest of the entries add their information about the traceability process into the SM. Ultimately, the customers can use the product ID (e.g., encrypted in a QR code) to look up the BSC smart contract. The output encompasses the entire journey of the product, starting from its initial stage as a seed, through the growth and transportation processes, until it reaches the supermarket. This information is guaranteed to be reliable due to the immutability of blockchain technology. In this paper, we use four participating entities (i.e., farmer, distributor, retailer, and consumer) to collect data of BSC for analysis. The details are described in Fig. 1.
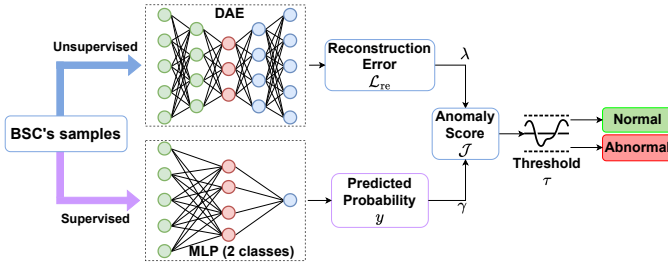
Fig. 2: Proposed semi-supervised DAE-MLP network for anomaly detection in BSCs.

B. Proposed Anomaly Detection Model

Fig. 1 shows our proposed system model for anomaly detection in BSCs. We assume that several participating entities and users interact with BSC through a blockchain end-point, i.e., Ethereum node. A detector is deployed at the top level of each Ethereum node. This detector comprises three components: a traffic monitor, a BC-ID tool, and our proposed DAE-MLP model. First, the traffic monitor captures the network traffic data from network interfaces of the Ethereum node via 'libpcap-dev' package. The captured traffic data consists of packets related to the network layer (e.g., ACK, SYN, etc.) and consensus and beyond layers (e.g., Ethereum tokens exchanges, BSCs' txs, etc.). Second, we use the BC-ID tool in [12] to extract 21 important features from these captured packets. At this stage, network packets are transformed into samples to be analyzed by DL networks. There are two phases in the cyber-attack detection of DL networks including offline training and online detection. During the offline training phase, both unlabeled and labeled samples, including normal and anomaly states, are used to train our proposed DAE-MLP network. In the online detection phase, the trained model is deployed into the detector. Thereby, it predicts anomalies in the extracted samples from the BC-ID tool. A key feature of our proposed DAE-MLP network is the ability to update the MLP component's hyper-parameters with newly recognized and labeled cyber-attacks, thereby enhancing the detector's accuracy.

III. PROPOSED DAE-MLP NETWORK

Our proposed semi-supervised DAE-MLP network is illustrated in Fig. 2. The network consists of two main components, i.e., DAE and MLP networks [13]. The DAE nework is an unsupervised NN characterized by a typically narrow central layer, referred to as the "bottleneck", which is designed to reconstruct the input data at the output. Specifically, it consists of two main modules, i.e., the encoder (E) and the decoder (D). The encoder compresses the input data into a lower-dimensional representation, capturing essential features while reducing noise and redundancy. The i-th layer of this module is expressed as follows:

$$\mathbf{z}(\mathbf{x})^{(i)} = f_{\mathrm{E}}^{(i)}\left(\mathbf{W}_{\mathrm{E}}^{(i)}\mathbf{x}^{(i-1)} + \mathbf{b}_{\mathrm{E}}^{(i)}\right), \qquad (1)$$

for $i = 1, \ldots, I$. At the first layer, $\mathbf{x}^{(0)} \in \mathbb{R}^n$ is an input sample with n features. $\mathbf{W}_{\mathrm{E}}^{(1)} \in \mathbb{R}^{m \times n}$ and $\mathbf{b}_{\mathrm{E}}^{(1)} \in \mathbb{R}^m$ are

weight matrix and bias vector of encoder module, respectively. $f_{\mathrm{E}}^{(i)}$ denotes the activation function, e.g., ReLU, Sigmoid, etc. The output $\mathbf{z}(\mathbf{x})^{(1)} \in \mathbb{R}^m$ is a latent representation which is utilized as the input of the next layer in this module. The decoder then reconstructs the latent space $\mathbf{z}(\mathbf{x})^{(I)}$ to form the output, which is expected the same as input. Each layer of the decoder often increases the number of neurons. The j-th layer of this module is expressed as follows:

$$\hat{\mathbf{x}}^{(j)} = f_{\mathrm{D}}^{(j)}\left(\mathbf{W}_{\mathrm{D}}^{(j)}\mathbf{z}(\mathbf{x})^{(j-1)} + \mathbf{b}_{\mathrm{D}}^{(j)}\right), \qquad (2)$$

for $j = 1, \ldots, J$. $\mathbf{W}_{\mathrm{D}} \in \mathbb{R}^{n \times m}$ and $\mathbf{b}_{\mathrm{D}} \in \mathbb{R}^n$ are weight matrix and bias vector of decoder module, respectively. The output $\hat{\mathbf{x}}^{(J)} \in \mathbb{R}^n$ is reconstructed version of $\mathbf{x}^{(0)}$. The difference between the input and output, also referred to as reconstruction error (re), can be evaluated by the mean squared error (MSE), as follows:

$$\mathcal{L}_{\mathrm{re}}(\mathbf{x}^{(0)}, \hat{\mathbf{x}}^{(J)}) = \frac{1}{n}\sum_{a=1}^{n}(x_a^{(0)} - \hat{x}_a^{(J)})^2. \qquad (3)$$

In our framework, the DAE is trained using normal traffic samples to learn the representation of the normal state. When network traffic containing cyber-attack samples is fed into the trained DAE model, the output significantly deviates from these inputs. This occurs because the distribution of abnormal samples differs from that of normal samples used during the DAE training process. Thus, $\mathcal{L}_{\mathrm{re}}(\mathbf{x}^{(0)}, \hat{\mathbf{x}}^{(J)})$ of abnormal states will be higher than the loss observed in the normal state. By setting a threshold (τ), we can effectively classify abnormal samples in network traffic of BSCs.

The latter component, a supervised MLP network (M), acts as a binary classifier (bi-classifier). The k-th layer, for $k = 1, \ldots, K$, of bi-classifier is given as follows:

$$\mathbf{y}(\mathbf{x})^{(k)} = f_{\mathrm{M}}^{(k)}\left(\mathbf{W}_{\mathrm{M}}^{(k)}\mathbf{x}^{(k-1)} + \mathbf{b}_{\mathrm{M}}^{(k)}\right), \qquad (4)$$

where notations and their shape resemble those of a layer in the encoder of the DAE network, as mentioned above. However, at the last layer of MLP component, we transform the input $\mathbf{x}^{(K-1)}$ into a scalar value instead of a vector, i.e., $0 \leq y^{(K)} \leq 1$. This value represents the predicted probability. If $y^{(K)} \geq 0.5$, the label is assigned as 1 (anomaly class); otherwise, the label is 0 (normal class).

The DAE-MLP then utilizes both reconstruction loss from the DAE component and predicted probability from the MLP component to form a total anomaly score ($\mathcal{J}$), as follows:

$$\mathcal{J} = \lambda \mathcal{L}_{\mathrm{re}} + \gamma y, \qquad (5)$$

where λ and γ are weights of reconstruction loss and predicted probability, respectively. Nonetheless, DAE and MLP networks have two different loss functions (i.e., MSE and binary cross-entropy, respectively) and generate distinct loss values in each training epoch. Hence, we combine two loss values in an epoch into a final loss value ($\mathcal{L}_{\mathrm{total}}$), to update hyper-parameters in both networks at once. This is achieved by weighting the losses with an α ratio.

$$\mathcal{L}_{\mathrm{total}} = \alpha \mathcal{L}_{\mathrm{DAE}} + (1 - \alpha)\mathcal{L}_{\mathrm{MLP}}. \qquad (6)$$

Algorithm 1 An efficiency threshold for the DAE-MLP.

Input: Initial threshold: $\tau \leftarrow \tau_{\text{init}}$
              Step size: $\zeta \leftarrow 0.001$
              Decay rate: $r \leftarrow 0.5$
              Decay times: $D \leftarrow 15$
              Number of worse accuracies in a row: $c \leftarrow 0$
              Maximum value of c: $c_{\text{max}} \leftarrow 10$
Output: An efficiency threshold: τ

1: **for** $d \leftarrow 1$ **to** D **do**
2:    **while** True **do**
3:        $\text{Acc}_d \leftarrow \texttt{evaluate\_model}(\lambda, \gamma, \tau)$
4:        $\tau \leftarrow \tau + \zeta$
5:        **if** $\text{Acc}_d > \text{Acc}_{\text{best}}$ **then**
6:            $\text{Acc}_{\text{best}} \leftarrow \text{Acc}_d$
7:            $c \leftarrow 0$
8:        **else**
9:            $c \leftarrow c + 1$
10:       **if** $c = c_{\text{max}}$ **then**
11:           $\tau = \tau - c_{\text{max}}\zeta$
12:           $\zeta \leftarrow r\zeta$
13:           $c \leftarrow 0$
14:           **break**

TABLE I: Architecture of the proposed DAE-MLP.

Component	Module	Layer	Input features	Output features	Activation function
DAE	Encoder	Linear	21	21	ReLU
		Linear	64	32	ReLU
		Linear	32	16	ReLU
	Decoder	Linear	16	32	ReLU
		Linear	32	64	ReLU
		Linear	64	21	Sigmoid
MLP	Classifier	Linear	21	32	ReLU
		Linear	32	16	ReLU
		Linear	16	1	Sigmoid

TABLE II: Descriptions of dataset and training parameters.

Parameter	Value	Parameter	Value
No. *Normal* samples	600,000	λ	0.5
No. *BP* samples	25,293	γ	0.5
No. *DoS* samples	100,000	α	0.5
No. *DoS_gas* samples	91,128	Train / test ratio	0.8/0.2
No. *OaU* samples	50,998	Learning rate	0.01
No. *FoT* samples	100,000	β	0.9

- Cyber-attacks in the network layer [12]:
  - *Brute-force Password (BP)*: Hackers target stealing EOA private keys of participating entities and users in BSCs using the BP attack.
  - *Denial of Service (DoS)*: Attackers send millions of network packets to blockchain nodes to increase the response time of BSC.
- Cyber-attacks in the consensus and beyond layers [2], [5]:
  - *DoS with Block Gas limit (DoS_gas)*: Functions inside the SM of BSCs could be temporarily unavailable if the required gas to execute these functions exceeds the gas limit. Hackers exploit this weakness to disrupt BSC's services.
  - *Overflow and Underflow (OaU)*: OaU vulnerabilities may be present in condition-checking statements in programming, potentially allowing hackers to bypass these checks and insert incorrect information into BSCs.
  - *Flooding of Transactions (FoT)*: Attackers send massive meaningless txs to blockchain nodes to increase the response time of BSC, similar to DoS in the network layer.

Based on the anomaly score, DAE-MLP decides whether the state of a sample, $A(\mathbf{x}, \lambda, \gamma, \tau, \alpha)$, is an anomaly or not by

$$A(\mathbf{x}, \lambda, \gamma, \tau, \alpha) = \left\{ \begin{array}{ll} \text{Normal}, & \text{if} \quad \mathcal{J} \geq \tau, \\ \text{Anomaly}, & \text{if} \quad \mathcal{J} < \tau. \end{array} \right. \quad (7)$$

As observed in Eq. (7), threshold plays an essential role in anomaly detection accuracy using DAE-MLP [10]. Therefore, we design the algorithm 1, which is our small proposal to find an efficiency threshold for a trained DAE-MLP model to detect anomalies in BSCs. This algorithm is based on the grid search method with a decay rate. Initially, the threshold (τ_{init}) value is a quantile function [14], given by:

$$\tau_{\text{init}} = Q_{\mathcal{L}_{\text{re}}\left(\mathbf{x}_{(\text{nor})}, \hat{\mathbf{x}}_{(\text{nor})}\right)}(\beta), \quad (8)$$

where β is the percentile of the quantile function. $\mathcal{L}_{\text{re}}\left(\mathbf{x}_{(\text{nor})}, \hat{\mathbf{x}}_{(\text{nor})}\right)$ is the reconstruction error of only normal state samples. The `evaluate_model` function returns the accuracy value of trained DAE-MLP with input arguments, i.e., λ, γ, and τ. Note that, after updating the MLP component, algorithm 1 must be executed again to adjust for the new hyper-parameters.

IV. EXPERIMENTAL RESULTS

A. Experiment Setup and Evaluation Methods

Real-world blockchain cyber-attacks aim at both peer-to-peer systems (i.e., network layer) and blockchain applications (i.e., consensus and beyond layers) [2]. These attacks caused various consequences such as delays in block confirmation time, loss of digital assets, and fake information from the past. In this paper, we perform experiments to replicate five types of cyber-attacks that can directly impact BSCs as follows:

In our laboratory, we launch an Ethereum network and a few servers as IoT gateways for generating datasets. The servers perform as participating entities in the BSC. They also act as users to update and track the information in this system. Additionally, we use another server to perform five types of cyber-attacks on the Ethereum network and the SM of the BSC. We then use the BC-ID tool in [12] to collect and extract features from network traffic of the blockchain nodes to create a dataset. Table I shows the architecture of our proposed DAE-MLP that will be used in the experiments in this paper. Table II provides descriptions of our collected dataset within the BSC system. This table also provides information about weights and other parameters during the DAE-MLP training process. In this paper, we use several metrics to evaluate models, i.e.,
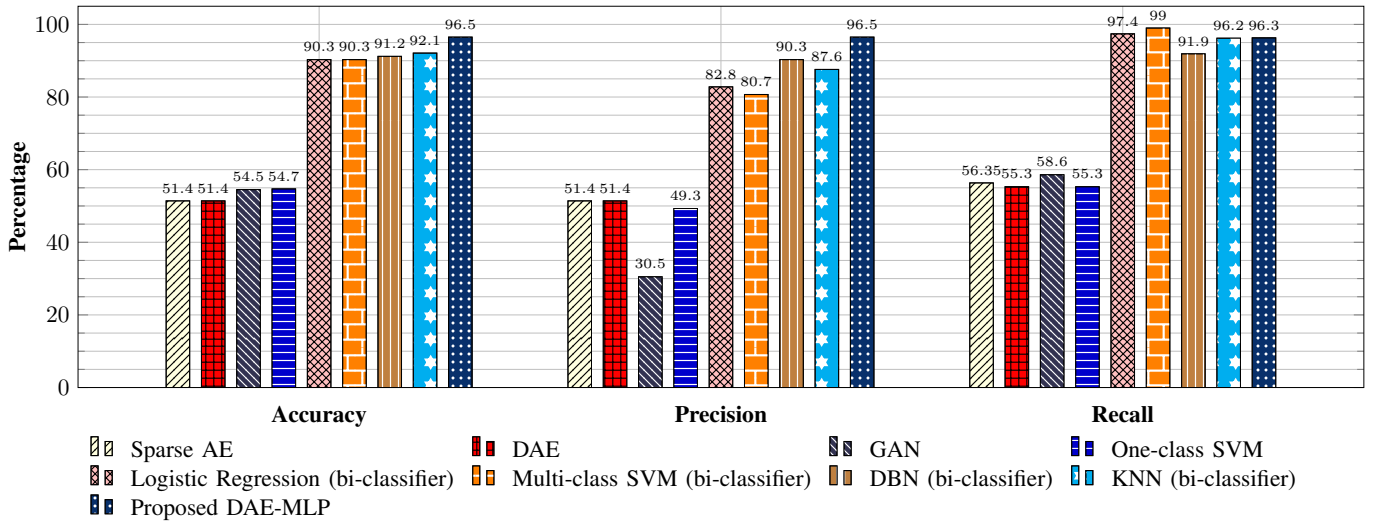
Fig. 3: The performance of proposed DAE-MLP vs. other anomaly detectors and bi-classifiers.

accuracy, F1-score, precision, and recall. These metrics are estimated using built-in functions from a widely-used machine learning library, i.e., scikit-learn [15]. Additionally, precision and recall metrics are calculated using the 'macro' averaging method to ensure fairness among classes with different sample sizes.

B. Performance Evaluation

1) Performance Comparison: Fig. 3 provides the performance results in accuracy, precision, and recall of our proposed DAE-MLP network and other models, such as unsupervised anomaly detectors, and several binary supervised models (bi-classifiers). In Fig. 3, we can see that the accuracies in cyber-attack detection of the unsupervised networks, i.e., Sparse DAE, DAE [16], GAN, and one-class SVM have low accuracy values, i.e., $51.4, 51.4, 54.5$, and 54.7, respectively. These results demonstrate that unsupervised networks are not effective in detecting cyber-attacks in the consensus and beyond layers by analyzing the network layer's packets in our dataset. Besides, supervised bi-classifiers, i.e., Logistic Regression, Multi-class SVM, DBN [12], and KNN, can achieve outstanding accuracy values, i.e., $90.3, 90.3, 91.2$, and 92.1, respectively. Based on these results, we can see considerable improvements in the accuracies from unsupervised anomaly detectors to bi-classifiers. However, our proposed (semi-) supervised DAE-MLP network can achieve performance in accuracy, precision, and recall of about $96.5, 96.5$, and 96.3, respectively. By employing the supervised component (i.e., MLP), the DAE-MLP can obtain higher performance results compared to other considered models Overall, the accuracy, precision, and recall of our proposed DAE-MLP network outperform other considered models for anomaly detection in our dataset.

2) Threshold Values: Table III provides the performance of DAE-MLP with different threshold values. We compare our proposed threshold value with those in [17]

TABLE III: Performance of proposed DAE-MLP with different threshold values.

τ	μ [17]	0.8μ	0.5μ	0.2μ	τ [18]	τ_{ours}
Accuracy	43.67	43.62	45.40	94.74	95.45	**96.52**
Precision	52.94	52.87	54.09	93.59	96.10	**96.47**
Recall	67.22	66.63	64.19	95.66	94.53	**96.29**

and [18]. In [17], the threshold is a value within the range from $\mathcal{L}_{\text{re}}\left(\mathbf{x}_{(\text{nor})}, \hat{\mathbf{x}}_{(\text{nor})}\right)$ to $\mathcal{L}_{\text{re}}\left(\mathbf{x}_{(\text{ano})}, \hat{\mathbf{x}}_{(\text{ano})}\right)$, where $\mathcal{L}_{\text{re}}\left(\mathbf{x}_{(\text{ano})}, \hat{\mathbf{x}}_{(\text{ano})}\right)$ is the reconstruction error of only anomaly state samples. Table III considers several values in this range, i.e., mean (μ), 0.8μ, 0.5μ, and 0.2μ. On the other hand, [18] proposed to used $\tau = Q_{\mathcal{L}_{\text{re}}\left(\mathbf{x}_{(\text{nor})}, \hat{\mathbf{x}}_{(\text{nor})}\right)}(0.9)$ as the threshold value. In this work, we use this value τ as the initial value τ_{init} for algorithm 1. In general, we can observe that our threshold value, τ_{ours}, gives more precision than other values with an accuracy of up to 96.5%. In contrast, the threshold values from [18] vary greatly, depending on the corresponding threshold values. Among these values, 0.8μ and 0.2μ are the worst and best threshold values with their accuracy, i.e., 43.62% and 94.74%, respectively. The reason can come from the significant overlap in the distribution of reconstruction errors between normal and anomaly state samples. Although the approach from [18] gives the acceptable accuracy of up to 95.45%, our proposed algorithm can enhance the performance of cyber-attack detection of the model. Specifically, with the initial threshold τ_{init}, our proposed algorithm can provide better performance in accuracy and recall than that of [18] by approximately 1% and 2%, respectively.

3) Adaptation: In this session, we simulate a practical scenario where attacks are not previously trained by the model. We focus on the capacity to adapt to novel cyber-attacks of DAE-MLP network within BSCs. In particular, we assume that the training dataset lacks a type of attack, e.g., BP attack.

TABLE IV: Accuracy detection of novel cyber-attacks before and after updating the MLP component.

Missing class	BP		DoS		DoS_gas		OaU		FoT	
Case	w/o	w	w/o	w	w/o	w	w/o	w	w/o	w
F1-score	74.03	79.39	60.98	94.08	86.08	91.80	81.68	86.79	88.70	93.18
Precision	64.18	66.74	79.69	88.82	86.56	87.85	78.18	79.92	88.10	89.02
Recall	87.46	97.95	49.38	100.0	85.60	96.11	85.51	94.96	89.30	97.76

However, this attack still appears while the model performs cyber-attack detection in BSCs. In this case, the model has to detect anomalies without being trained by them. After that, we record the anomaly and update the training dataset to improve the accuracy of cyber-attack detection. Table IV shows performance in F1-score, precision, and recall of the model when being trained without a type of attack (w/o) and being updated with the attack (w). In general, as in Table IV, without updating new attacks, the performance of the model reduces little with all types of attacks. In detail, our proposed model can detect FoT in the case of w/o with 88.7% and 88.1% in F1-score and precision, respectively. The results are slightly lower than the case of w with 93.18% and 89.02% in F1-score and precision, respectively. Besides, the proposed model can detect DoS attacks in the case of w/o with an F1-score of 60.98% and precision of 79.69%. In the case of w, the F1-score is 33.1% higher and the precision is 9.13% higher than the case of w/o, resulting in an F1-score of 94.08% and precision of 88.82%. As a result, our proposed model can not only detect anomalies without previously being trained by the types of attacks but also improve the performance of detecting unknown attacks by allowing the model to update the detected cyber-attacks.

V. CONCLUSION

In this paper, we proposed a novel semi-supervised learning framework for anomaly detection in BSCs. Specifically, we first implemented an Ethereum network in our laboratory to collect a dataset for blockchain-based supply chains, which includes normal and attack traffic data. After that, we designed DAE-MLP which is a semi-supervised model that can leverage the detection performance of the unsupervised DAE model by combining it with the supervised MLP model. Additionally, a grid search algorithm was employed to find the efficient threshold, thereby improving anomaly detection accuracy in the considered model. The simulation results showed that our proposed model could have better performance in detecting anomalies than other supervised and unsupervised approaches. Moreover, the proposed model can not only detect anomalies without learning the type of attack but also improve the cyber-attack detection performance after updating new attacks to the model. Regarding future work, we aim to study more types of anomalies and design more intelligent models to enhance the accuracy of anomaly detection in BSCs.

REFERENCES

[1] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1676–1717, Dec. 2018.

[2] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, "Exploring the attack surface of blockchain: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1977–2008, Mar. 2020.

[3] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar, "Blockchain-based soybean traceability in agricultural supply chain," *IEEE Access*, vol. 7, pp. 73 295–73 305, May 2019.

[4] A. Vangala, A. K. Das, N. Kumar, and M. Alazab, "Smart secure sensing for iot-based agriculture: Blockchain perspective," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17 591–17 607, Aug. 2021.

[5] J. Cheng, L. Xie, X. Tang, N. Xiong, and B. Liu, "A survey of security threats and defense on blockchain," *Multimedia Tools and Applications*, vol. 80, pp. 30 623–30 652, Aug. 2021.

[6] M. Asante, G. Epiphaniou, C. Maple, H. Al-Khateeb, M. Bottarelli, and K. Z. Ghafoor, "Distributed ledger technologies in supply chain security management: A comprehensive survey," *IEEE Transactions on Engineering Management*, vol. 70, no. 2, pp. 713–739, Feb. 2023.

[7] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. CRC Press, Apr. 2016.

[8] P. F. de Araujo-Filho, M. Naili, G. Kaddoum, E. T. Fapi, and Z. Zhu, "Unsupervised gan-based intrusion detection system using temporal convolutional networks and self-attention," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4951–4963, Dec. 2023.

[9] S. Dong, Y. Xia, and T. Peng, "Network abnormal traffic detection model based on semi-supervised deep reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4197–4212, Dec. 2021.

[10] J. Kim, M. Nakashima, W. Fan, S. Wuthier, X. Zhou, I. Kim, and S.-Y. Chang, "A machine learning approach to anomaly detection based on traffic monitoring for secure blockchain networking," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3619–3632, Sept. 2022.

[11] Y. K. Sanjalawe and S. R. Al-E'mari, "Abnormal transactions detection in the ethereum network using semi-supervised generative adversarial networks," *IEEE Access*, vol. 11, pp. 98 516–98 531, Sept. 2023.

[12] T. V. Khoa, D. H. Son, D. T. Hoang, N. L. Trung, T. T. T. Quynh, D. N. Nguyen, N. V. Ha, and E. Dutkiewicz, "Collaborative learning for cyberattack detection in blockchain networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 7, pp. 3920–3933, Apr. 2024.

[13] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.

[14] R. J. Hyndman and Y. Fan, "Sample quantiles in statistical packages," *The American Statistician*, vol. 50, no. 4, pp. 361–365, Mar. 1996.

[15] Scikit-learn, "Metrics and scoring: quantifying the quality of predictions," Accessed: Apr. 14, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html

[16] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, Mar. 2020.

[17] N. Najari, S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia, "Radon: Robust autoencoder for unsupervised anomaly detection," in *14th International Conference on Security of Information and Networks (SIN)*, Dec. 2021.

[18] H. Homayouni, S. Ghosh, I. Ray, S. Gondalia, J. Duggan, and M. G. Kahn, "An autocorrelation-based lstm-autoencoder for anomaly detection on time-series data," in *IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 5068–5077.